

**UNITED STATES PATENT APPLICATION**

*of*

**Yassir K. Elley**

**Anne H. Anderson**

**Stephen R. Hanna**

**Sean J. Mullan**

*and*

**Radia J. Perlman**

*for a*

**METHOD AND SYSTEM FOR PRO-ACTIVE CREDENTIAL  
REFRESHING**

00547183-044000

# METHOD AND SYSTEM FOR PRO-ACTIVE CREDENTIAL GATHERING

## RELATED CASES

This application discloses subject matter also disclosed in the following pending  
5 applications, filed herewith and assigned to Sun Microsystems, Inc., the assignee of this  
invention:

U.S. Patent Application entitled "METHOD AND SYSTEM FOR  
PRESENTATION OF NON-REVOCATION CERTIFICATES" (Atty. Docket No.  
P3314/RSH/112047-0005), filed on May 10, 1999 and given a serial number 09/307,953;

10 U.S. Patent Application entitled "METHOD AND SYSTEM FOR DYNAMIC  
ISSUANCE OF GROUP CERTIFICATES" (Atty. Docket No. P3685/RSH/112047-  
0012), filed on May 10, 1999, whose serial number has yet to be assigned; and

U.S. Patent Application entitled "METHOD AND SYSTEM FOR PROVING  
MEMBERSHIP IN A NESTED GROUP USING CHAINS OF CREDENTIALS" (Atty.  
15 Docket No. P3686/RSH/112047-0013), filed on May 10, 1999, whose serial number has  
yet to be assigned.

## FIELD OF THE INVENTION

This invention relates generally to authorization for access to a resource between  
entities in a network, and more particularly to the maintenance of credentials required for  
20 access to the resource.

## BACKGROUND OF THE INVENTION

During ordinary operation of computer networks it is usual for a client to access a  
server and to request access to a resource provided by that server. A client may be

thought of as a program running on a work station, desktop type computer, personal digital assistant (PDA) or even an embedded device, and a server may be thought of as a program performing a service for a plurality of clients. The client may also be thought of as the computer running the client software, and the server may also be thought of as the computer running the server software. For some purposes, the client may be thought of as a user on whose behalf a request is being made. In some cases, the same computer may run both the client software and the server software. The service is ordinarily provided by the execution of a server program at the request of the client. Specifically, the service provides a resource to the client. The resource may be any operation that is executed, affected or controlled by a computer, such as a word processing or spread-sheet program, the transfer of files, or some other data processing function. The resource access may also include the ability to read or to modify entries in a database, execute or modify a program maintained by the server, or even modify data maintained by another computer in the system.

In deciding whether or not to grant access to a resource, a resource server must answer two questions:

A. "Is the client correctly identifying himself?" and

B. "Is the identified client authorized to access the requested resource?"

The first question involves a process called "client authentication." The second involves reference to an authorization decision mechanism, such as an Access Control List (ACL) maintained by the server and containing a list of individual clients and/or client groups who are permitted access to the resource.

Client authentication can be accomplished using public key cryptographic methods, as described in *Network Security, Private Communication in a Public World*, Charlie Kaufman, Radia J. Perlman, and Mike Speciner, PTR Prentice Hall, Englewood Cliffs, New Jersey, 1995, (Kaufman et al.) at chapters 5, and 7 and 8, pages 129-161 and 177-222. Specifically, client Alice can authenticate herself to resource server Bob if she knows her private key and Bob knows Alice's public key. Bob has obtained Alice's public key in an identity certificate from a trusted certification authority or from a certifica-

tion authority in a chain extending from a trusted authority. Other methods of authentication may be used and the present invention does not depend on which method is used.

An identity certificate may be revoked. One common method of dealing with revocation involves the use of Certificate Revocation Lists (CRLs) which are analogous  
5 to the books of revoked credit card numbers that were at one time published and distributed periodically to merchants. Like these books, CRLs suffer from being expensive to distribute and are therefore infrequently distributed. There may also be a significant period of time between certificate revocation and CRL distribution, during which the resource server is unaware of the revocation.

10 For maximum security, the certificate authority may be off-line and therefore inaccessible on a transaction-by-transaction basis. Moreover, issuance of an identity certificate may be a relatively lengthy process so that, even if the certificate authority is online, it is impractical to issue an up-to-date certificate for each transaction. An alternative approach to certificate revocation involves the use of on-line revocation servers, which  
15 maintain lists of revoked identity certificates. With on-line revocation servers, up-to-date revocation status can be determined.

At the same time, if a revocation server's private key has been compromised, the damage will be more limited than if an on-line certification authority's private key were compromised. Specifically, if the certification authority's private key were compromised,  
20 the authority might issue new certificates to unauthorized clients. On the other hand, a compromised revocation server would result only in continued access by a client with revoked authorization. Specifically, a compromised revocation server can never grant unauthorized access to a client who has never had authorized access. Although a compromised revocation server may wrongly revoke an authorized client, the revocation  
25 would only be a denial-of-service attack.

The use of on-line revocation servers, which is analogous to the method employed today for the authorization of credit card purchases, is also expensive because the resource server usually contacts an on-line revocation server at each transaction to determine whether the certificate has been revoked. The OCSP (On-line Certificate Status

Protocol) proposed standard of the PKIX working group (RFC 2560) at  
<ftp://ftp.isi.edu/in-notes/rfc2560.txt>, specifies that the revocation status for each cer-  
tificate can be retrieved from the revocation server and cached by the resource server  
verifying that certificate. Although caching improves resource server efficiency, it still  
5 places a burden on the resource server, which may already be burdened with the process-  
ing of resource access requests.

An authentication and authorization arrangement introduced by the Open Soft-  
ware Foundation (OSF) and known as the Distributed Computing Environment (DCE)  
model has a central database on a machine known as a "privilege server" or "central  
10 trusted authority." When a client logs on to the system, the privilege server issues a se-  
cret, or symmetric, key certificate (as opposed to a public, or asymmetric, key certificate)  
identifying all the groups of which the client is a member. The client presents this certifi-  
cate to any server on which the client wishes to access a resource. The resource server  
has an ACL for the resource, and the ACL includes both authorized clients and client  
15 groups. If neither the client nor any one of the groups of which the client is a member is  
listed in the ACL, client access is denied. This approach saves some work for the server,  
but requires that a central trusted authority know all the groups of which the client is a  
member and also that the client's group list is small enough so that presentation of the en-  
tire collection is not unwieldy. The DCE model is described in Kaufman et al. at Section  
20 17.7, pages 455-459.

When a client wishes to access a resource on a server, at least one of them needs  
to gather and maintain various credentials for client identification and authorization.  
These credentials may include public key identity credentials, group membership creden-  
tials, group non-membership credentials and non-revocation credentials. In some cases,  
25 the client gathers the various credentials and submits them to the server for access to the  
resource. However, for the client, such task may be time consuming and particularly  
burdensome in situations where the gathering occurs at the time of access. The client  
may therefore cache the credentials for future use. However, the server may have various  
constraints on the credentials it will accept, such as recency requirements. If the server

refuses to accept “stale” credentials, the client may have to re-gather the credentials, thereby delaying the access. In other cases, the server does the work of gathering the various credentials to determine whether a client is authorized to access the resources. If the server gathers the various credentials at the time of client access, this may cause an undesirable delay to the client. The server may also cache the various credentials. However, if the cached credentials do not meet the server’s recency requirements, it will have to re-gather the credentials, thereby delaying the access. This problem is not limited to the client-server situation; it exists whenever an entity needs to gather credentials and keep them up to date.

10

## SUMMARY OF THE INVENTION

In accordance with the invention, the entity that has the burden of gathering the credentials, pro-actively refreshes the credentials and keeps them current. For instance, a presenter of the credentials, for example, a client, pro-actively refreshes the credentials such that at the time of presentation, the credentials meet the resource-specific constraints of a recipient of credentials, for example, a resource server. Generally, these certificates include time stamps designating the date and time of issue. For each resource that it protects, a resource server typically establishes various constraints such as a recency requirement which specifies how recently a credential has to have been issued to be accepted an adequate credential. Other constraints may include maximum certificate chain length, trust level and so forth. In another instance, a recipient of credentials pro-actively gathers and refreshes credentials to prevent un-authorized access to the various resources it is protecting.

According to an embodiment of the invention, the entity that has the burden of maintaining the credentials, stores in a memory, a profile collection that contains one or more credential profiles. A credential profile includes information such as a credential identifier, when the credential was issued, when the credential was last used, on which resource, resource constraints such as a recency requirement and so forth. At a refresh trigger event, a refresh procedure is performed in accordance with a refresh policy. For

example, the refresh policy may be simple such as (1) refresh all credentials older than 10 minutes, or it may have more than one criterion such as (2) refresh all credentials used in the last 24 hours and older than 10 minutes, or (3) refresh all credentials used in the last 24 hours and only if older than the resource recency requirement. These various criteria  
5 may be deduced from the information stored in the credential profiles. Note that the refresh policies are merely examples. The refresh policies will be tailored according to a desired result.

Using a refresh policy, the entity maintaining the profile collection examines each credential profile in the profile collection to determine if it needs to be refreshed. If the  
10 credential needs to be refreshed, then the entity pro-actively refreshes the credential. Once the credential has been updated in the credential profile, the entity proceeds to the next credential profile and repeats the process until all the credentials in the profile collection have been examined.

The pro-active refreshing of credentials is not limited to the client-server situation,  
15 but applies to any entity that presents credentials (including chains of group credentials) to another entity. For example, an entity may send a signed e-mail to a recipient and may also present credentials that the recipient may find useful in validating the signature on the e-mail. For a given transmission, the entity presenting credentials is defined as a “presenter of credentials”, and the entity receiving the credentials is termed as a “recipient of credentials”. A particular entity may be a presenter of credentials in one transmission and a recipient of credentials in a second transmission. For example, Alice and Bob may want to mutually authenticate each other for communications between them, in which case, each would have to present credentials to the other. When Alice transmits credentials to Bob, Alice is a presenter of credentials and Bob is a recipient of credentials.  
20 Alternatively, when Bob transmits credentials to Alice, Bob is a presenter of credentials and Alice is a recipient of credentials. Other aspects and advantages of the invention will be further described in the detailed description to follow below.

## BRIEF DESCRIPTION OF THE DRAWINGS

The above and further advantages of the invention may be better understood by referring to the following description in conjunction with the accompanying drawings in which like reference numbers indicate identical or functionally similar elements:

- 5        Fig. 1 is a block diagram of a computer network;
- Fig. 2 is an example of an Access Control List (ACL);
- Fig. 3 is an example of a non-revocation certificate;
- Fig. 4 is a flow diagram of a credential gathering and processing procedure; and
- Fig. 5 is a flow diagram of a pro-active credential refreshing procedure.

## 10        DETAILED DESCRIPTION OF AN ILLUSTRATIVE EMBODIMENT

### Introduction

The basic concept is that when a resource is to be accessed, the entity that has the burden of gathering the credentials, has an up-to-date set of credentials. This is accomplished by pro-actively refreshing the credentials so as to keep them current. In one instance, a presenter of credentials, for example, a client, pro-actively refreshes its credentials such that at the time of presentation, the credentials meet the resource-specific constraints of a recipient of credentials, for example, a resource server. Generally, these certificates include time stamps designating the date and time of issue. For each resource that it protects, a resource server typically establishes various constraints such as a re-  
15        cency requirement, which specifies how recently a credential has to have been issued to be accepted as an adequate credential. Other constraints may include maximum certificate chain length, trust level and so forth. In another instance, a recipient of credentials pro-actively gathers and refreshes credentials to prevent un-authorized access to the various resources it is protecting.

25        Each group has a name and the location of its group server can be obtained from a system directory. The group also has a public key and an associated private key, which is used for signing group certificates. Further, a group can have as members individuals and other groups. That is, a "parent", or nested, group may have "child" groups for members.



It may even have a complicated Boolean expression for membership, such as  $G1 \rightarrow (G2 \text{ AND } G3) \text{ AND NOT } (G4)$ , meaning that all members of groups G2 and G3 are in the group G1, except for members of group G4.

A group membership or group non-membership certificate usually indicates membership status for a specified name, e.g., client "Alice" is a member of group G1, although the certificate may also indicate membership status for a specified public key or other identity. In most cases, a group membership list associated with a group will be maintained by an on-line group server. However, the on-line group server may need to dynamically decide whether a given client is a member of the group, i.e. the server may do more than simply refer to a membership list. In the case where there is no group membership list, group membership will be determined by some other criterion. For example, group membership could be determined by a particular attribute of a client. In another case, where the client may be a member of a child group, the on-line group server may obtain proof of the client's membership in the child group. Alternatively, the client may present the proof of membership in the child group to the on-line group server.

A fresh group membership certificate can be obtained from an on-line group server at any time and presentation of the certificate to the resource server will be sufficient to prove membership in the group. However, an off-line group server infrequently issues certificates, such as when a client joins the group or when the entire membership list is updated. The client will then request a newly-issued non-revocation certificate from an associated on-line revocation server and present it, along with the group membership certificate, to the resource server.

#### *Group Memberships and Nested Groups*

As discussed above, a group may have as its members other groups, or subgroups and the client may contact child group servers to obtain proof of membership in a child group. For example, the aforementioned client Alice may attempt to access a resource on resource server "Bob". If Alice is not listed as an individual on the resource ACL, but groups G1, G2 and G3 are listed on the ACL, Bob returns to Alice the message:

"Access denied, unless you can prove membership in group G1, G2 or G3."

In contrast to traditional systems, this message does not have to be sent during session establishment (i.e., the initial handshake between the client and the server). It may be the case that Alice had previously established the session with Bob and at this later time has decided to access the resource protected by Bob. At this point Bob may challenge Alice  
5 to present additional credentials. Alice may have recently obtained a membership certificate for one of these “root” groups in the course of obtaining access to some other server or for some other reason. If not, the client system can prompt the human operator to provide guidance as to the groups of which the client is likely to be a member. This may save substantial time if there are a large number of groups on the ACL.

10 If human intervention is not desirable, an exhaustive search may be undertaken: Alice communicates with on-line group servers containing the group membership lists of groups G1, G2 and G3 and attempts to obtain a membership certificate from one of these servers.

In another variation, rather than performing an exhaustive search, Alice may be  
15 able to narrow the search by relying on previously stored information to determine groups in which Alice is likely to be a member. For example, if Alice has an old group membership certificate for group G1, Alice can attempt to obtain a new group membership certificate from the G1 group server before undertaking an exhaustive search for a group certificate.

20 Although Alice may not be listed as a member of group G1, group G1 may be a nested group, i.e., it may be a parent to the child groups, or subgroups, G5 and G6. The G1 group server will ask Alice:

“Can you prove membership in group G5 or G6?”

Alice then communicates with the G5 and G6 servers. For example, if group G5 lists  
25 Alice as a member, the G5 server returns a group membership certificate in group G5. Alice returns to the G1 server to request a group membership certificate, armed with the certificate from the G5 server. The G1 server then grants Alice a group membership certificate. Now, armed with the certificate from root group G1, Alice can go to Bob and obtain access to the requested resource. A group membership certificate is not the only

mechanism by which Alice may prove membership in group G5 to the G1 server. For example, Alice may alternatively present the group G5 membership list, signed by group G5, or some other proof of group membership.

The example presented above is rather simple. In some cases, Alice will be unable to establish membership in either a root or a child group and will be denied access to the resource. In other cases, Alice may have to search down several subgroup levels before finding membership in a group. To facilitate this task, Alice maintains a family tree for each root group, tracing the path of subgroups "visited" during a search. Alice can then easily detect and abort loops (where G5 is a member of G1 is a member of G5 is a member of G1 and so on). When membership in a subgroup is found, Alice moves back up the path collecting a group membership certificate from each successive group server and presenting it to the next higher group server until the root group is reached. Alice then presents to Bob the group membership certificate issued by the root group server.

In the above scenario, it has been assumed that the group servers issuing group membership certificates are on-line and thus create newly-issued group certificates at runtime. In that case, the group membership certificates created at runtime are fresh enough so as not to need any further proof of non-revocation. If a group certificate is "old", i.e., it was obtained by the client more than some specified time prior to a request for access to a resource, as ascertained from a time stamp included in the certificate, the resource server will require a newly-issued certificate from the on-line group server.

If the root group server is off-line, the client may obtain from an on-line source, such as a directory, a certificate stating that the client, or a subgroup, is a member of the root group. The client will then request a corresponding non-revocation certificate from an on-line revocation server. In the above example, if the G1 server is off-line, Alice will assure Bob that the certificate designating group G5 as a member of group G1 has not been revoked, i.e. Alice will retrieve a newly-issued non-revocation certificate from the group G1 on-line revocation server. When the client is a member of a subgroup and the root group server is off-line, the client will present a chain of two or more group membership certificates to the resource server. Additionally, the client will present a valid non-

revocation certificate from the appropriate on-line revocation server for the group membership certificate issued by the off-line root group server. If the next lower group server is also off-line, the client will also present a valid non-revocation certificate from the appropriate on-line revocation server for the group membership certificate issued by the off-

5 line subgroup group server. The chain continues until it includes either a group membership certificate issued by the highest level on-line group server in the family tree, or the group membership and non-revocation certificates for all subgroups down to the lowest level group in the family tree. In the present example, Alice will present to Bob the certificate designating group G5 as a member of group G1 and its corresponding non-

10 revocation certificate, and will also present a group membership certificate issued by the on-line G5 server designating Alice as a member of group G5. Similarly, if a group server is on-line, but its subgroup servers are off-line, Alice may need to present a chain of group membership certificates to the on-line group server in order to prove membership in a subgroup.

15 In a different example, the G1 group server may grant membership to anyone who can prove membership in group G11 and non-membership in group G12. Accordingly, Alice will retrieve a group membership certificate from the G11 group server and a group non-membership certificate from the G12 group server and present those certificates to the G1 group server. The G1 group server will then issue a G1 group membership certificate, which Alice will present to Bob. If the G1 group server is off-line, Alice will

20 need to present to Bob two chains of credentials, including proofs of group membership and non-membership. Specifically, Alice will need to present to Bob a first chain including a group G1 membership list (signed by G1), along with the group membership certificate from group G11, and a second chain including the signed group G1 membership

25 list, along with the group non-membership certificate from group G12. In this case, because Alice does not need to twice present the signed group G1 membership list, Alice simply presents the signed group G1 membership list, along with the group G11 membership certificate and the group G12 non-membership certificate. Thus, in order to prove

membership in a nested group, Alice will sometimes need to present both proofs of group membership and proofs of group non-membership.

### *Group Non-membership Certificates*

A resource server may also prohibit access to a resource based on client membership in one or more groups. In this case, the client will gather and present group non-membership certificates stating that the client is not a member of the designated groups. For example, group G1 members may be permitted access to a resource, unless they are also group G2 members. Alice will have to prove both membership in group G1 and NON-membership in group G2. To prove non-membership in group G2, Alice will present a group G2 non-membership certificate to Bob. Alice requests a non-membership certificate from the G2 group server and presents the certificate, along with a group G1 membership certificate to Bob.

The work required to gather the credentials necessary to prove group non-membership is more intensive than that required for group membership. For each prohibited root group, the client will be required to prove non-membership in each and every group extending from the root. For example, Bob may deny resource access to all members of group G2. Therefore, Alice will request a group non-membership certificate from the G2 server. The root group G2 might have as members the child groups, or subgroups, G7 and G8. The G2 group server will ask Alice:

“Can you prove non-membership in groups G7 and G8?”

Alice then requests a group non-membership certificate from both the G7 and G8 servers. If group G7 also lists the groups G9 and G10, Alice requests a group non-membership certificate from both the G9 and G10 servers. Alice presents the G9 and G10 group non-membership certificates to the G7 server which then issues a group non-membership certificate. Alice next presents the G7 and G8 group non-membership certificates to the G2 server and receives a G2 group non-membership certificate. Now, armed with a group non-membership certificate from group G2, Alice can go to Bob and prove non-membership in root group G2.

The case of off-line group servers becomes more difficult when trying to prove non-membership in a nested group. For example, if in the above example the G2, G7, and G8 group servers are off-line, Alice will need to present a chain of credentials to Bob, including proofs of both group membership and group non-membership. Specifically,  
5 Alice may present to Bob a group G2 membership list (signed by G2), a group G7 membership list (signed by G7) and a group G8 membership list (signed by G8). From these credentials, Bob will be aware of the memberships of groups G2, G7 and G8, and Bob will be able to indirectly verify that Alice is not a member of any of these groups. In particular, Bob will know that group G7 lists groups G9 and G10 as members. Therefore, in  
10 order to prove non-membership in group G7, Alice will also need to present to Bob group non-membership certificates for the groups G9 and G10.

#### An Embodiment of the Invention

As shown in Fig. 1, a computer network 100 includes a network "cloud" 102 that provides the interconnection for devices on the network. The network cloud 102 may  
15 represent a simple local area network, for example, an Ethernet on one floor of a building. At the other extreme, it may represent the entire worldwide Internet. The network cloud 102 may contain transmission lines, repeaters, routers, network backbones, network interconnect points, etc., depending upon the extent of the network which it represents.

A client can be any device capable of sending messages over the network and is  
20 generally thought of as an individual workstation, a desk-top computer, a mini-computer accessed by a terminal, a personal digital assistant (PDA), an embedded device, or some other relatively simple computer. A client is often a computer operated by one person, although an independently operating computer or a program operating without human intervention can also be a client. Client computer Alice 104 and two additional client  
25 computers 106, 108 are shown connected to the network cloud 102. A modern network may include thousands of client computers.

A resource server Bob 110 is also connected to network cloud 102. A resource server can be any device capable of receiving messages over a network and is usually thought of as a larger computer, which contains resources to which client computers de-

sire access. For example, a resource may be a database, a file system, etc. A resource 112 on resource server Bob 110 represents any resource to which a client may desire access. An Access Control List (ACL) 114 contains a list of clients which are permitted to access the resource 112. As a convenience, clients may be assigned membership in  
5 groups of clients, designated groups G1, G2, G3, ..., GN, having associated group servers 130, 132, 134, 136. Accordingly, ACL 114 may also contain the names of groups whose member clients are permitted access to the resource 112.

An Off-line Certification Authority (OCA) server 120 issues identity certificates used by clients to identify themselves when seeking access to various resources on vari-  
10 ous servers, such as client Alice 104 access to resource 112. A switch 122 represents the ability of the OCA server 120 to be temporarily connected to the network cloud 102 so that it may, at selected times, issue an identity certificate to a client. The switch 122 is in "open" position most of the time to protect the OCA server 120 from attacks by malicious persons.

15 An On-line Revocation (OR) server 124 is connected to network cloud 102 on a substantially permanent basis. The OR server 124, upon request from a client, issues a non-revocation certificate stating that a particular client's identity certificate, previously issued by the OCA server 120, has not been revoked as of the time stamp. The non-revocation certificate is then transmitted to the requesting client.

20 Fig. 2 shows a typical Access Control List (ACL) 200 having a name field 202, in this case "112", and access entries. The first access entry 204 specifies that client Alice 104 is permitted access. Additional access entries for client computers x1 and x2 205, 206, along with groups G1, G2 and GN 208, 210, 212, round out the list.

25 Fig. 3 shows a typical non-revocation certificate 300 issued to client Alice 104 by the OR server 124. Client Alice 104 had previously obtained a certificate from the OCA server 120. The OR server 124 maintains a list of certificates which have been revoked. Upon receipt of a request from a client, the OR server 124 checks its revocation list and, assuming that the subject certificate is not on that list, issues a non-revocation certificate. The first entry 302 in the non-revocation certificate 300 indicates that a previously issued

certificate for client Alice 104 has not been revoked. Additionally, the non-revocation certificate 300 includes a signature entry 304, and a time stamp comprising an issue date entry 306 and time entry 308.

Resources may have recency requirements for credentials, such as non-revocation  
5 certificates, group membership certificates and group non-membership certificates. For example, resource server Bob 110 may require that the credentials used to access the resource 112 be no more than one-day old, or possibly no more than 10 minutes old, depending upon the level of security desired for the resource 112, the number of clients requesting the resource 112, and the number of requests which OR server 124 can handle.

10 Before describing a pro-active credential refreshing procedure, an illustrative credential profile gathering and processing procedure will first be described as shown in Fig. 4. For each procedure, each entity on the computer network may include a processor with an associated memory, which contains instructions for performing one or more steps of the procedure. Persistent storage of these instructions may be in a server system remote  
15 from the network entity and its processor. It is understood that the procedure performed by instructions may be stored in a magnetic or optical disks, semiconductor memories and the like, collectively referred to as a processor executable medium. The electrical signals that carry digital data representing the instructions are exemplary forms of carrier waves used for transporting information from a server system to a network entity.

20 Referring to Fig. 4, at block 401, client Alice 104 needs a particular credential. At block 402, Alice 104 determines if there is a credential profile for the needed credential. In one example, a credential profile is a record stored in the memory and includes information such as; a credential identifier, when the credential was issued, when the credential was last used, on which resource, resource constraints such as a recency requirement and so forth. From the credential profiles, Alice 104 retrieves the necessary credentials  
25 required to access resources. A profile collection is, as the name implies, a collection of the credential profiles. Below is an example of a profile collection:



CREDENTIAL	ISSUED	LAST USED	RESOURCE	REGENCY REQUIREMENT
G1 GROUP CERT	9 AM 10/2/99	10 AM 10/4/99	SUN'S WEB	1 HOUR
NON-REVOCATION CERT	10 AM 11/5/99	10 AM 12/6/99	ENG SPEC	20 MINUTES

If Alice 104 does not detect a credential profile for the needed credential, then at  
5 block 404, Alice 104 creates an empty credential profile. At block 406, Alice proceeds to  
gather the credential and related information that make up the credential profile.

In one example, the credential and related information may be obtained from  
manual entry by the user including an administrator. In another example, Alice 104  
“learns” the resources accessed by the user over a period of time and predicts which cre-  
10 dentials are required for those resources. For instance, Bob 110 may control several re-  
sources that are accessed by the user. When the user accesses one or more of Bob’s re-  
sources, Alice 104 creates the credential profiles having resource-specific information on  
those accessed resources. For example, Alice 104 may learn that Bob 110 requires a  
chain of group certificates before allowing access to a certain resource. Obtaining a chain  
15 of group certificates is generally time consuming. Thus, Alice 104 retrieves and stores  
those group certificates, along with its corresponding resource-specific information in a  
credential profile. Alice 104 may deduce the various resource-specific information from  
various access attempts. In instances where it is available, Alice 104 may directly obtain  
the resource-specific information from a directory or from Bob.

20 Blocks 408-414 are generally the steps performed by Alice 104 for credential  
processing where she retrieves a credential for resource access. In block 408 (which is  
optional) Alice 104 may check the recency requirement of a credential in a credential pro-

file to determine if the credential is fresh (i.e., if the credential meets the resource server's recency requirement). If not, then Alice 104 initiates a credential refresh procedure (such as the example shown in Fig. 5) to gather and store a fresh credential. Otherwise, in block 410, Alice 104 retrieves the credential from the credential profile for resource access. In block 412, Alice 104 updates the field that indicates the certificate last used in the credential profile. In block 414, Alice uses the retrieved credentials to access the resource.

So far, we have assumed that Alice 104 is the one that will be doing the work to gather the necessary credentials and to create a profile collection. It may be that Alice 104 may not have the capacity to perform the work because she does not have sufficient processing power. Otherwise, it may be that the resource server Bob 110 may not want to divulge various resource-specific information to Alice 104 for security reasons. Thus, in the case where Bob 110 does the work, Bob similarly performs the procedure above to gather and process the necessary credentials. It is assumed that Bob 110 knows the resource-specific constraints for the resources that he is protecting. Thus, referring to Fig. 4, in block 402, when client Alice 104 requests access to one of Bob's resources, Bob 110 looks for a credential profile for the needed credential in his memory to determine whether Alice 104 is allowed access. If the credential profile does not exist, in block 404, Bob 110 creates an empty credential profile. At block 406, Bob proceeds to gather the credential that makes up the credential profile. Because Bob 110 knows its resource-specific constraints, Bob will probably be able to insert the various constraints in the profile along with the gathered credential. For example, Bob 110 may notice that he needs a group membership list from group server G1 to determine whether access should be granted to a particular resource, and from the resource constraints, Bob may know that the recency requirement is one hour from the time the list is issued. Thus, Bob 110 may refresh the group member list every hour during his spare time and allow Alice 104 to access the resource if she is a member in the list. Blocks 408-414 are generally the steps performed by Bob 110 for the credential processing and is similar to that performed by Alice, which has been described above.

Fig. 5 is an illustrative flow diagram 500 of a pro-active credential refreshing procedure. As described above, there are many instances in which resource server Bob 110 will not accept from client Alice 104, certificates that are not current. In those instances, previously, Alice 104 requested new credentials at the time of access, which delayed the Alice-Bob transaction. The problem is further compounded in instances where Alice 104 needs to prove a group membership involving a chain of group membership certificates or a complicated Boolean construction of group membership certificates. The pro-active refreshing procedure keeps the credentials fresh and available for resource access.

In block 502, client Alice 104 detects a trigger event that initiates the execution of the refresh procedure. The refresh procedure may be performed in accordance with a refresh policy that has been previously predetermined. For example, the refresh policy may be simple such as (1) refresh all credentials older than 10 minutes, or it may have more than one criteria such as (2) refresh all credentials used in the last 24 hours and older than 10 minutes, or (3) refresh all credentials used in the last 24 hours and only if older than the resource recency requirements. These various criteria may be deduced from the credential profiles stored with the credentials. Note that the above mentioned refresh policies are merely examples and the refresh policies are tailored according to a desired result.

In blocks 504-506, Alice 104 examines a credential profile in the profile collection to determine if it needs to be refreshed. For example, assuming refresh policy (3) is used for refreshing purposes, Alice 104 first checks the pertinent field in the credential profile to determine if the credential was last used within the 24 hours. If so, then Alice 104 compares the recency requirement with the credential issue date, both which are also stored in the credential profile. If the credential is determined to be stale, in block 508, Alice 104 pro-actively refreshes the credential. Once the credential has been refreshed, in block 510, Alice updates the issue date of the credential in the profile. Alice 104 may sequentially examine the various credential profiles in the profile collection until the credentials are fresh as performed by the repetition routine of blocks 512-514.

Note that there are instances in which Alice 104 may not refresh credentials that are stale. For example, Bob 110 may have a recency requirement of 24 hours on credentials that are presented for access to a certain resource. However, Alice 104 may know that she needs to present those credentials every Friday afternoon for access and, thus, may refresh the credentials on Friday morning so that they will be fresh at access time. In another instance, Alice 104 may keep track of how often the various resources are accessed and further, when the various resources are accessed. With this information, she predicts with certain probability when the various resources will be used and refreshes the credentials such that the credentials are fresh at access time. There may be instances where a certain resource is rarely used, thus, Alice 104 may decide that it is too time consuming to refresh the credentials of that resource and decide to gather the credentials at access time. In yet another instance, Alice 104 may retrieve previously used credentials before any resource access is performed. For example, if yesterday, client Alice used group membership certificates for G1 and G2 to access various resources, she stores the certificates in the memory. When Alice 104 logs in today she may obtain group membership certificates from the memory before accessing any resources since there is a likelihood that she will need them.

In the case where the resource server Bob 110 is doing the work, Bob may also perform a similar refreshing procedure as shown in Fig. 5. In block 502, Bob 110 detects a trigger event that indicates a performance of a refresh procedure. The trigger event may be in accordance with a refresh policy that has been previously determined. In blocks 504-506, Bob 110 examines a credential profile in the profile collection to determine if it needs to be refreshed. If the credential needs to be refreshed, in block 508, Bob 110 proactively refreshes the credential. Once the credential has been refreshed, in block 510, Bob updates the issue date of the credential in the profile. Bob 110 may sequentially examine the various credential profiles in the profile collection until the credentials are fresh as performed by the repetition routine of blocks 512-514.

The foregoing description has also been directed to specific embodiments of this invention. It will be apparent, however, that other variations and modifications may be

1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2350 2351 2352 2353 2354 2355 2356 2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388 2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2440 2441 2442 2443 2444 2445 2446 2447 2448 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2479 2480 2481 2482 2483 2484 2485 2486 2487 2488 2489 2490 2491 2492 2493 2494 2495 2496 2497 2498 2499 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2515 2516 2517 2518 2519 2520 2521 2522 2523 2524 2525 2526 2527 2528 2529 2530 2531 2532 2533 2534 2535 2536 2537 2538 2539 2540 2541 2542 2543 2544 2545 2546 2547 2548 2549 2550 2551 2552 2553 2554 2555 2556 2557 2558 2559 2560 2561 2562 2563 2564 2565 2566 2567 2568 2569 2570 2571 2572 2573 2574 2575 2576 2577 2578 2579 2580 2581 2582 2583 2584 2585 2586 2587 2588 2589 2590 2591 2592 2593 2594 2595 2596 2597 2598 2599 2600 2601 2602 2603 2604 2605 2606 2607 2608 2609 2610 2611 2612 2613 2614 2615 2616 2617 2618 2619 2620 2621 2622 2623 2624 2625 2626 2627 2628 2629 2630 2631 2632 2633 2634 2635 2636 2637 2638 2639 2640 2641 2642 2643 2644 2645 2646 2647 2648 2649 2650 2651 2652 2653 2654 2655 2656 2657 2658 2659 2660 2661 2662 2663 2664 2665 2666 2667 2668 2669 2670 2671 2672 2673 2674 2675 2676 2677 2678 2679 2680 2681 2682 2683 2684 2685 2686 2687 2688 2689 2690 2691 2692 2693 2694 2695 2696 2697 2698 2699 2700 2701 2702 2703 2704 2705 2706 2707 2708 2709 2710 2711 2712 2713 2714 2715 2716 2717 2718 2719 2720 2721 2722 2723 2724 2725 2726 2727 2728 2729 2730 2731 2732 2733 2734 2735 2736 2737 2738 2739 2740 2741 2742 2743 2744 2745 2746 2747 2748 2749 2750 2751 2752 2753 2754 2755 2756 2757 2758 2759 2760 2761 2762 2763 2764 2765 2766 2767 2768 2769 2770 2771 2772 2773 2774 2775 2776 2777 2778 2779 2780 2781 2782 2783 2784 2785 2786 2787 2788 2789 2790 2791 2792 2793 2794 2795